

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



**Grado en Ingeniería en Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

**Tecnologías WHMS (Wearable Health Monitoring Systems):
Análisis e integración de datos de un sensor inercial.**

**Julián Miguel Pérez Dones
Tutor: Javier Garrido Salas**

Junio 2019

Tecnologías WHMS (Wearable Health Monitoring Systems): Análisis e integración de datos de un sensor inercial

AUTOR: Julián Miguel Pérez Dones

TUTOR: Javier Garrido Salas

**Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2019**



Resumen (castellano)

Este Trabajo Fin de Grado tiene como objetivo obtener, analizar e integrar los datos obtenidos de un sensor inercial, que nos sirva de punto de partida para el diseño y fabricación un “dispositivo vestible”, más conocido por el término en inglés wearable, que nos permita, mediante la utilización de un conjunto de tres IMU (unidad de medida inercial), junto con un microprocesador embebido, medir y analizar los cambios en la posición de la espalda de una persona y detectar una incorrecta posición, lanzando un mensaje de alerta para que el usuario la corrija.

El proceso se inicia con el análisis de las medidas de una sola IMU conectadas a una placa de desarrollo Arduino, para que una vez comprendido el funcionamiento de la misma, proceder a integrar las otras dos obteniéndose el wearable final.

Es decir, tras analizar y entender los datos medidos, utilizaremos directamente el microcontrolador y la IMU, sin depender de la placa de desarrollo, para finalmente obtener la PCB final.

Abstract (English)

This Bachelor Thesis consists on obtain, analyze and integrate the data obtained from an inertial sensor which will be the starting point for the design, and manufacture of a wearable that allows us, by using a set of three IMU together with an embedded microprocessor, to measure and analyze changes on the back position of the user. The system launches an alert message to the user too to correct it.

We will start by analyzing the measurements of a single IMU connected to and Arduino development board, then after understanding the results, integrate the other two inertial sensors to obtain the final wearable.

After analyzing and understanding the measured data, we will directly use the microcontroller and IMU's without depending on the development board, to finally obtain three PCB.

Palabras clave (Castellano)

Wearable, IMU, microprocesador, postura corporal, PCB, embebido, DMP, ángulos de Euler, cuaterniones, sensor inercial.

Keywords (Inglés)

Wearable, IMU, microprocessor, PCB, embedded, digital motion processor, Euler angles, quaternions, inertial sensor.

Agradecimientos

A mis padres, porque desde el primer día que inicié esta aventura han sabido aconsejarme dirigirme, y apoyarme, porque son el pilar en el que me apoyo y siempre me han aconsejado lo mejor.

A mi novia, porque siempre que te he necesitado has estado ahí, en los días bueno pero sobretodo en los malos, por siempre saber calmarme cuando solo tú podías hacerlo, por apoyarme y quererme como lo haces.

A mis hermanos, Diego y Lucía, mis compañeros de aventuras de puertas para afuera, por teneros siempre cerca aunque estuvierais lejos.

A mis compañeros de clase, mis amigos de siempre, mis compañeros de trabajo, a mis profesores y a todos los que han pasado por mi vida durante estos años, gracias por llevarme hasta aquí y hacerme ser la persona que soy hoy en día.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
	Organización de la memoria.....	3
2	Estado del arte	4
2.1	Era digital, tecnológica, e inteligencia de las cosas.....	4
2.2	Trabajos previos, antecedentes	4
2.3	Unidad de Medida Inercial, estructura y componentes	5
2.3.1	Acelerómetro	6
2.3.2	Giróscopo	7
2.4	Aplicaciones de las IMU	9
3	Arquitectura del sistema	10
3.1	Idea final del proyecto.	10
3.2	Diseño en placa de desarrollo Arduino.....	13
3.3	Diseño en PCB final	15
4	Desarrollo	20
4.1	Desarrollo en Arduino	20
4.1.1	Análisis previo, datos en raw o DMP.	20
4.1.2	Obtención de los ángulos de Euler	22
4.1.3	Representación del movimiento en tiempo real	23
4.2	Desarrollo con microcontrolador en protoboard	27
4.2.1	Subsección Problemas y aprendizaje.....	27
4.3	PCB final	28
4.3.1	Subsección Problemas y aprendizaje.....	29
5	Conclusiones y trabajo futuro.....	30
5.1	Conclusiones.....	30
5.2	Trabajo futuro	31
	Referencias	32
	Anexos.....	XXXV
A	Ángulos de Euler vs cuaterniones	XXXV
B	Filtro de Madgwick	XXXVIII
C	Código Arduino utilizado:	- 1 -
D	Código matlab utilizado para la representación de los ángulos de Euler:	- 3 -

INDICE DE FIGURAS

FIGURA 1 DISPOSITIVOS SELECCIONADOS: SENSOR MPU-9250 INCORPORADO A UNA PCB Y MICROCONTROLADOR AT MEGA32U4.....	11
FIGURA 2 ESQUEMA DEL FLUJO DE DISEÑO.....	12
FIGURA 3 CONEXIÓN FÍSICA DE LOS DISPOSITIVOS.....	15
FIGURA 4 ESQUEMÁTICO DE LA PLACA PROMICRO DE ARDUINO EXTRAÍDA DE [17].....	16
FIGURA 5 CONEXIÓN FÍSICA DE LOS DISPOSITIVOS ALIMENTADOS POR USB.....	16
FIGURA 6 ESQUEMÁTICO DE LA PLACA PROMICRO DE ARDUINO EXTRAÍDA DE [17].....	18
FIGURA 7 ESQUEMÁTICO DEL ENRUTAMIENTO DE LA PLACA	18
FIGURA 8 ATMEGA32U4 Y MPU 9250 SOLDADOS EN LA NUEVA PCB.....	19
FIGURA 9 POSIBLE IDEA DE PRESENTACIÓN FINAL	19
FIGURA 10 APROXIMACIÓN AL CÁLCULO DE LOS ÁNGULOS DE ROTACIÓN MEDIANTE ACELERACIONES.	22
FIGURA 11 REPRESENTACIÓN DE LOS ÁNGULOS DE NAVEGACIÓN.	26
FIGURA 12 ENRUTAMIENTO DE LA CAPA SUPERIOR.....	28
FIGURA 13 ENRUTAMIENTO DE LA CAPA INFERIOR.....	28

INDICE DE TABLAS

TABLA 1 TABLA DE ESPECIFICACIONES COMUNES DE UN ACELERÓMETRO.....	6
TABLA 2 TABLA DE ESPECIFICACIONES COMUNES DE UN GIRÓSCOPO BASADA EN TABLA DE [10]...	7
TABLA 3 CRITERIOS APLICADOS PARA LA SELECCIÓN DEL SENSOR.	11
TABLA 4 CONEXIONES ENTRE LA IMU Y LA PLACA DE DESARROLLO MEDIANTE I2C.	14
TABLA 5 DIRECCIONAMIENTO DEL I2C	14
TABLA 6 ESPECIFICACIONES DEL CHIP SPARKFUN PRO-MICRO.	15

1 Introducción

1.1 Motivación

Utilizar las grandes ventajas que nos proporcionan las nuevas tecnologías para favorecer la mejora del bienestar físico de las personas es un tema interesante y sobre el que vienen centrándose un amplio número de investigaciones y aplicaciones actuales. Desde esta perspectiva, cuando reflexioné sobre el tema sobre el que realizaría mi TFG, mi inclinación por intentar crear algo de utilidad en este ámbito resultó determinante.

La evidencia de que en la actualidad un gran número de personas realiza su jornada laboral delante de un ordenador, prolongando el tiempo en que permanece sentada, podría generar una exposición a posturas de espalda forzadas e inadecuadas que podrían ser una fuente de dolencias. [1]

Pero no solo este grupo de personas serán de relevancia para nosotros. Según estudios recientes del “REIDE (Red Española de Investigadores en Dolencias de Espalda)” junto a la fundación Kors, las dolencias lumbares o en general de la espalda son cada vez más comunes en personas mayores, embarazadas incluso jóvenes en edad escolar. [2]

Atendiendo a estas reflexiones, el campo de aplicación del trabajo planteado se nos ampliaba al conjunto de la población que potencialmente podría sufrir de problemas de espalda por no adaptarse a su posición adecuada.

En definitiva, la motivación inicial de nuestro proyecto es conseguir un sistema capaz de detectar esta incorrecta postura de la espalda, habitual cuando los individuos permanecen sentados durante periodos prolongados de tiempo, para así ayudar a subsanar o prevenir problemas de espalda, una dolencia que presenta una tendencia creciente.

Para el logro de este objetivo global, nos planteamos que el uso de una unidad de medida inercial o IMU (Inertial Measurement Unit), se presentaba como una metodología adecuada. Por ello, y como parte en la consecución del mismo, se propone este TFG, que consistirá en el análisis detallado del funcionamiento de este sensor hasta conseguir su total comprensión, para por último, integrarlo en una PCB de tamaño reducido.

Esto servirá como punto de arranque para el desarrollo de un producto final, un wearable formado por 3 IMU, que se utilizará para medir la postura corporal relativa de un sujeto para, en caso de una postura incorrecta o un movimiento demasiado brusco de la espalda poder lanzar un mensaje de alerta.

1.2 Objetivos

El objetivo de este trabajo será el análisis y la integración de datos obtenidos mediante un sensor inercial.

Para poder alcanzar este objetivo final es preciso abordar previamente unos objetivos intermedios, que nos permitirán comprender plenamente el funcionamiento de estos sensores, un instrumental que a día de hoy están presentes en múltiples sistemas y nuevos desarrollos.

El primer paso abordado se centró en la comprensión y el correcto análisis de los datos que podíamos obtener de la IMU utilizando una placa de desarrollo. En un principio suponíamos que este estudio no sería de gran complejidad debido a que este tipo de sensores están a la orden del día, admitiendo que encontraríamos documentación al respecto. Así mismo al realizar este primer análisis sobre una placa de desarrollo Arduino presumimos que existirían numerosas librerías de código para llevar a cabo nuestra tarea, pero según se fue avanzando en el análisis y la obtención de estos datos, descubrimos numerosas dificultades y algunos problemas que describiremos a lo largo de esta memoria.

A partir de esta incidencia, abordamos los siguientes pasos de una forma que, el conocimiento adquirido sobre este tipo de sensores fuera lo más profundo posible para poder posteriormente avanzar en nuestro desarrollo sin depender de librerías de terceros, y conseguir acercarnos a un teórico dispositivo final “wearable”.

De esta manera nuestro siguiente propósito acometido se centró en cubrir un objetivo fundamental para otorgar de viabilidad a nuestra propuesta. Concretamente afrontamos la necesidad de eliminar la dependencia de la placa de desarrollo Arduino, ya que esta placa es de gran tamaño y no nos serviría de cara a nuestro objetivo final de realizar un wearable, que necesariamente debe poderse llevar encima por sus potenciales usuarios de forma cotidiana.

Finalmente, para alcanzar el propósito de este trabajo, se ha creado nuestra PCB uniendo el sensor y el microcontrolador, cerrando con ello el objetivo de crear nuestro prototipo final, un instrumento capaz de dar respuesta a una necesidad creciente y potencialmente válida para la mejora de la calidad de vida de las personas.

Organización de la memoria

La memoria consta de los siguientes capítulos:

- Estado del arte. En este capítulo se pretende contextualizar la situación tecnológica actual. Nuestro deseo es conocer en qué punto nos encontramos respecto a la utilización de sensores electrónicos, más concretamente inerciales y, para ello, se presentan algunas aplicaciones previas de este tipo de sensores, tanto de fuera como de dentro del ámbito de nuestro estudio. Adicionalmente, dentro de esta sección encontraremos un pequeño marco teórico en el que se describirá, brevemente, la composición de la IMU y el funcionamiento de los sensores que ésta lleva incorporados.
- Arquitectura del sistema. En este apartado describiremos paso a paso nuestro esquema de proyecto. Se irán describiendo el proceso de creación del prototipo, partiendo de una primera idea inicial, presentando las ejecuciones parciales y sus dificultades hasta alcanzar el proyecto final que da contenido a este TFG. En esta sección, nos centraremos fundamentalmente en la exposición de los sensores elegidos, y el porqué de esta elección, una decisión que ha resultado fundamental para la definición completa de la arquitectura del sistema propuesto.
- Desarrollo. El grueso del trabajo y quizás la parte más comprometida y las aportaciones más personales de este trabajo dan contenido a esta sección. En ella se describirán cada uno de los pasos seguidos hasta llegar al proyecto resultante final. Es decir, se incorporan de forma detallada tanto los errores cometidos en cada una de las etapas que han sido cubiertas en el proceso de creación de la IMU, como el aprendizaje y la consolidación de habilidades obtenidos en cada una de ellas.
- Integración, pruebas y resultados. Como no podría ser de otro modo, la verificación de la validez del sistema construido conlleva varias fases de contrastación y, por ello, en este apartado se da cabida a la presentación de las actuaciones de control sobre el proceso de integración de las IMUS, las pruebas que definen situaciones múltiples de posicionamiento y la sensibilidad del prototipo creado. Es decir, en este bloque se analizan los resultados obtenidos en cada parte experimental, de forma que el análisis del sistema y los contrastes de validez de resultados, nos permitan concluir con una valoración positiva sobre el nivel de resolución de los objetivos alcanzados.
- Conclusiones y trabajo futuro. Por último, se indicarán posibles trabajos a posteriori partiendo de la base de este proyecto. Líneas que se podrían seguir para continuar este proyecto y mejorarlo. Además, se extraerán las ideas principales del desarrollo y se valorará el aprendizaje que ha supuesto este trabajo.

2 Estado del arte

2.1 Era digital, tecnológica, e inteligencia de las cosas

Vivimos en una época tecnológica. La tecnología llegó hace ya algunos años a nuestras vidas y lo hizo para quedarse, e instaurarse hasta en los objetos más cotidianos.

Desde hace algo más de una década, los términos aprendizaje automático (machine learning), o el internet de las cosas IoT han cobrado una especial relevancia. La tendencia del ser humano está siendo la de dotar a los objetos de la mayor independencia posible, de humanizar las cosas. Por ello sectores como la robótica, la electrónica y la informática se han desarrollado exponencialmente en los últimos años, apareciendo nuevos dispositivos como las pulseras y relojes inteligentes y algunos años antes los smartphones o los drones, los cuales están cada vez más presentes en nuestro día a día.

Esta gran evolución o revolución tecnológica, esta humanización de las cosas es posible gracias en gran medida a los sensores electrónicos. Los sensores electrónicos están presentes hoy en día en casi cualquier parte, sin diferenciación de sectores ni de utilidades, haciendo que su estudio e implantación en nuevos desarrollos y aplicaciones sea una realidad.

Existen centenares de sensores, y se multiplican exponencialmente no sólo ante la disponibilidad de un volumen de datos muy grande sobre los que realizar mediciones, sino ante la necesidad de disponer de distintos parámetros para las diferentes aplicaciones que van surgiendo según avanza la tecnología.

Podríamos hablar de cada uno de los distintos sensores electrónicos que existen en la actualidad, pero sería una labor demasiado extensa, fuera del alcance de este trabajo, por tanto, nos centraremos en los sensores que utilizaremos en el mismo, y presentaremos las distintas aplicaciones que estos sensores tienen la actualidad.

2.2 Trabajos previos, antecedentes

En el año 2003, se diseña un sistema mediante el cual, a partir de la integración de las señales obtenidas de un giróscopo, se miden en tiempo real los movimientos de la espina dorsal. Partiendo de una posición inicial de cada sujeto, se realizan tres experimentos: flexión, extensión y rotación de la columna vertebral, de donde se obtuvo una base de datos que fue graficada y analizada a través de estudio de sus desviaciones típicas. [3]

Diez años más tarde, en 2013, se presenta un novedoso sistema de seguimiento y aproximación inercial basado en un filtro recursivo con constantes integradas. Este filtro permite hacer una estimación detallada y eficiente en tiempo real de los movimientos de la espina dorsal. Para ello el sistema divide la columna vertebral en 3 segmentos con 3 puntos equidistantes entre sí en cada uno de ellos, que actúan como las bases para la obtención de la información. De hecho, para la obtención de los datos necesarios el sistema utiliza transformaciones DH asumiendo una calibración inicial respecto a la IMU (0), situada en la parte del coxis nos da la posición y orientación del resto de IMUs. Además, para calibrar el sistema, se utiliza un filtro extendido de Kalman. [4]

En 2014 se propone un nuevo modelo para la medición y control de la pisada del usuario basado en la utilización de IMUs. En este caso tan solo se utilizan los datos obtenidos del giróscopo y acelerómetro, ya que se considera que la orientación del pie será siempre la misma [5]

Ya en el año 2016 un estudio proponía la utilización de las IMU para medir los ángulos de inclinación de cinco puntos distribuidos a lo largo de una línea. El algoritmo utilizado posteriormente está basado en las curvas cuadráticas de Bézier definidas respecto del primer punto de la anterior línea, es decir respecto del primer sensor.

Las curvas se obtienen a partir del conocimiento de las distancias entre los sensores y las medidas de los ángulos de inclinación tomados por las IMU.

Aplicando este planteamiento al ámbito que nos ocupa, la columna se reconstruye entonces con cuatro segmentos de las curvas de Bézier generadas, haciendo hincapié en la continuidad de las tangentes.

$$B(t) = (1 - t)^2 P_0 + 2t(1 - t)P_1 + t^2 P_2 \quad t \in [0,1]$$

Ecuación 2.1- Ecuación de la curva de Bézier.

Las posturas obtenidas se comparan con una postura patrón adecuada, que ha sido identificada por un médico especialista, a través del cálculo de la media de distancias absolutas entre ambas.

En aquellas situaciones en las que esta diferencia se aleja de los valores promedio, superando los umbrales admitidos, la utilización de vibradores enganchados a la ropa del usuario transmitirán un aviso, en tiempo real, de la existencia de una mala posición de su espalda.

Con todo ello el sistema completo quedaría formado por [6]:

- Una pieza de ropa con las IMU, vibradores y un controlador.
- Sistema de comunicación móvil.

2.3 Unidad de Medida Inercial, estructura y componentes

El sensor en torno al cual gira nuestro estudio es la IMU.

Hoy en día las unidades de medida inercial son sensores cada vez más utilizados que han ido aumentando su presencia en aplicaciones muy diversas y en campos heterogéneos que van desde la robótica, pasando por la aviación, e incluso en ciertas aplicaciones deportivas. Estos chips generalmente están compuestos de tres sensores, acelerómetro, giróscopo y magnetómetro, aunque en ocasiones este último puede no encontrarse, aplicándose cada uno de ellos a los tres ejes cartesianos, encontrando por tanto dispositivos con 6 o 9 grados de libertad (6 o 9 DOF) dependiendo si incluye o no magnetómetro.

La aplicación principal de estos sensores es la obtención de la posición, orientación y velocidad de un dispositivo. Para ello se utilizan los datos obtenidos por los tres sensores anteriormente mencionados y se combinan para obtener las medidas finales.

Para entender cómo obtener las medidas deseadas a partir de los datos que nos dan los sensores que contiene la IMU debemos entender primero como trabaja cada uno de ellos por separado.

2.3.1 Acelerómetro

Un acelerómetro es un sensor inercial que, como su nombre indica, mide aceleraciones gravitatorias, o lo que es lo mismo la tasa de variación de la velocidad. Pueden medir esta variación en dos unidades diferentes. O bien en la unidad del sistema internacional para la aceleración, es decir en (m/s^2) o bien en fuerza-G (g). Para nosotros en la superficie terrestre, una fuerza G [g] es equivalente a $9,8 \text{ m/s}^2$ aunque esto puede variar significativamente en función de la altitud [7].

En la siguiente tabla mostraremos algunos de parámetros más relevantes a la hora de seleccionar un acelerómetro en dos de los sectores donde este tipo de sensor es más utilizado.

Tabla 1 Tabla de especificaciones comunes de un acelerómetro

Parámetro	Automoción	Navegación
Rango	$\pm 2\text{g}$	$\pm 1\text{g}$
Rango de frecuencia	400 Hz-DC	100 Hz-DC
Resolución	$<10\text{mg}$	$<4\mu\text{g}$
Sensibilidad	$<5\%$	$<0.1\%$
Rango de temperatura	$-40^\circ\text{C}-85^\circ\text{C}$	$-40^\circ\text{C}-85^\circ\text{C}$

¿Cómo funciona un acelerómetro?

Los acelerómetros son dispositivos electromagnéticos capaces de medir tanto fuerzas estáticas como dinámicas. Dentro de las fuerzas estáticas encontramos la fuerza de la gravedad, mientras que en las fuerzas dinámicas encontramos vibraciones y movimiento.

Pueden medir aceleraciones en 1, 2 o 3 ejes, aunque hoy en días los de 3 ejes son los más comunes debido a que su coste de desarrollo es menor.

La mayoría de los acelerómetros funcionan de la misma forma. Están compuestos internamente por placas capacitivas, las cuales están unidas a pequeños muelles o resortes. Cuando una fuerza de aceleración actúa sobre el sensor, estos pequeños resortes se mueven internamente, haciendo que la capacitancia entre las placas varíe. A partir de los cambios en esta capacitancia se determina la aceleración. [8]

Además, físicamente un acelerómetro puede ser modelado como un sistema de segundo orden: masa, amortiguador, muelle, donde encontramos una masa “m” y la constante elástica del muelle “k”. A partir de la ley de Hook, junto con la segunda ley de Newton, se podrían medir las aceleraciones gravitatorias del sistema. [8], [9].

2.3.2 Gir6scopo

Los gir6scopos son sensores capaces de medir el movimiento rotacional sobre su eje. Son sensores, peque1os, generalmente baratos que miden la velocidad angular a la que son sometidos. Las unidades de medida son grados/segundo ($^{\circ}/s$) o bien revoluciones por segundo. En t6rminos generales podemos decir que los gir6scopos miden la velocidad de rotaci3n del dispositivo que los contiene.

La resoluci3n, la derivada asociada, o la salida en estado estacionario son algunos de los principales problemas que definen el rendimiento de estos sensores.

Tabla 2 Tabla de especificaciones comunes de un gir6scopo basada en tabla de [10].

Par6metro	Tasa de graduaci3n	Graduaci3n t6ctica	Graduaci3n inercial
Deriva[$^{\circ}/h$]	10 -1000	0.1 -10	<0.01
Precisi3n factor de escala [%]	0.1-1	0.01-0.1	<0.001
Rango de escala completa[$^{\circ}/s$]	50-1000	>500	>400
Ancho de banda[Hz]	>70	~100	~100

¿C3mo funciona un gir6scopo?

Al igual que los aceler3metros, los gir6scopos son capaces de medir velocidades angulares en los 3 ejes cartesianos y tambi3n comparten su existencia en tres versiones. Es decir, aunque existen gir6scopos que funcionan solo en 1 o dos ejes, son los que actúan sobre los 3 ejes los m6s populares en la actualidad, debido a su tama1o peque1o y precio reducido [10].

Para detectar la velocidad angular del dispositivo un gir6scopo utiliza el desplazamiento de una peque1a masa resonante que contiene en su interior. Este movimiento se transforma en peque1as se1ales el3ctricas que posteriormente son amplificadas y tratadas por un microcontrolador.

Sin embargo, no basta con conocer la velocidad angular para el logro de nuestro estudio. No es posible alcanzarlo si no se consigue medir la orientaci3n respecto a un ángulo inicial conocido. Por tanto, es necesario integrar los datos de la velocidad angular, obtenidos con el gir6scopo, con el sensor en el tiempo, tomando la variaci3n de la velocidad angular en intervalos de tiempo muy peque1os.

No obstante, con la integraci3n de estos valores a1un no se dispone de un ángulo total de giro, sino que obtenemos un ángulo para cada instante de tiempo, haciendo necesario realizar un proceso acumulativo, una suma de las variaciones y una media posterior para llegar a conocer el ángulo deseado. Este algoritmo de suma introduce una derivada, se ve acumulando a lo largo del tiempo y exige un proceso de correcci3n para que nuestras medidas sean lo m6s precisas y reales posibles.

2.3.3 Magnetómetro

Los magnetómetros son dispositivos capaces de medir campos magnéticos. En concreto miden densidad de flujo magnético. Su unidad es el Tesla (T) aunque algunos magnetómetros nos dan la densidad de flujo magnético en (As/m^2).

Los magnetómetros miden la densidad de flujo magnético en el punto del espacio donde esté situado el sensor. Esto es de vital importancia ya que al estar rodeados de distintos campos magnéticos, las medidas del magnetómetro pueden verse modificadas por distintos factores tales como la cantidad de instrumentos electrónicos situados cerca del sensor, la cantidad de metales cercanos, la situación geográfica donde nos encontremos etc... es por ello que debemos calibrar el sensor adecuándolo a la ubicación en donde nos encontremos antes de realizar medidas con él.

¿Cómo funciona un magnetómetro?

Hay distintos tipos de magnetómetros, y cada uno se basa en distintos principios para obtener sus medidas. Sin embargo, y dado a que en nuestro proyecto no será de vital importancia el cómo obtener estas medidas, no profundizaremos en cómo estos instrumentos las obtienen, para concentrar nuestros esfuerzos en definir con más detalle el funcionamiento general de los sensores y en los criterios y características que han sido aplicados para seleccionar los que se han considerado idóneos para nuestro proyecto.

Entre las características con las que se identifican los diferentes sensores caben ser citadas:

-Rango: El rango de nuestros sensores será la diferencia entre la máxima y la mínima medida que es capaz de otorgar cada sensor. Dependiendo de nuestra aplicación debemos seleccionar un sensor que se adecúe en rango a nuestras necesidades.

-Precisión: es la diferencia entre dos medidas consecutivas que puede dar nuestro sensor, es decir la diferencia mínima entre dos valores distintos que puede medir.

-Error o Bias: Cualquier sensor contendrá un error en sus medidas. Dependiendo del fabricante y de la calidad del sensor este error puede llegar a ser muy significativo y debemos tenerlo siempre en cuenta.

En nuestro caso concreto, puesto que no utilizaremos estos sensores por separado, es decir, con carácter autónomo o individual, sino que los utilizaremos integrados en una IMU, y nuestra aplicación no requiere en un primer desarrollo de una extrema precisión en las medidas, a la hora de seleccionar nuestro sensor, no se ha sido especialmente exigente, aunque si se han considerado niveles aceptables en esta trilogía de parámetros.

Una vez conocida cómo funcionan, qué miden y que parámetros son importantes, a la hora de seleccionar una IMU, nos cuestionamos ¿qué aplicaciones tienen?, ¿por qué cada vez se oye más el término IMU?, ¿en qué sectores se están utilizando? Y buscando respuestas hemos desarrollado el siguiente epígrafe.

2.4 Aplicaciones de las IMU

Hace más de 20 años que estos sensores aparecieron en el mercado. De aquellas primeras IMU a las actuales podemos observar grandes variaciones en su precisión, tamaño, velocidad... pero sus principios fundamentales de funcionamiento siguen siendo similares.

Debido a que la principal función de una IMU es medir variaciones en la posición, velocidad y aceleración de un cuerpo las aplicaciones de este tipo de sensores son infinitas. Desde su creación hasta la actualidad los principales sectores en los que se utilizan este tipo de sensores son automoción, transporte, comunicaciones, defensa, e industria, aunque las IMU han aumentado su presencia en otras muchas y diferentes áreas de conocimiento y de actividad.

Sobre el conjunto de aplicaciones sectoriales, hemos localizado diversas y múltiples utilidades de estos dispositivos, aunque se concentran en número y potencialidad en el ámbito de la robótica, la navegación y los radares.

No obstante, la innovación en sus aplicaciones se ha ido extendiendo, tanto es así que en 2015 se diseña un sistema basado en un filtro no lineal de Kalman, junto con un sistema MEMS-IMU, un sistema micro electro mecánico, y un sistema GPS, sistema de posicionamiento global, para medir el rendimiento de diferentes deportistas durante sus entrenamientos [11], una aplicación innovadora a la que se unen otras muchas vinculadas al campo de la navegación.

Así, estos sensores están presentes en todos los vehículos, desde los coches, trenes, helicópteros y aviones comerciales hasta los vehículos militares más sofisticados. Es por ello que numerosos estudios y proyectos se centran en la mejora de los sistemas de navegación integrados en este tipo de vehículos. Para ello tratan de mejorar la precisión de los mismos, minimizar sus errores, compensar sus derivas... entre otras muchas mejoras. [12], [13], [14].

Pero también las IMU son parte fundamental en el mundo de la robótica. En un mundo cada vez más robotizado las posibilidades y la funcionalidad que estos sensores ofrecen para el desarrollo de este sector son de vital importancia. Más concretamente dentro del área de los drones estos sensores inerciales forman parte del núcleo de estos dispositivos, dado que, en la gran mayoría de los casos, son los responsables de mantener en equilibrio, y estabilizar el dron durante su funcionamiento. La mejora en la autonomía y control de estos aparatos ha permitido que sus aplicaciones sean cada vez más variadas. Por ejemplo, un reciente proyecto de la UPM [15] ha integrado estos sensores inerciales junto con un sistema GPS en una flota de drones para la vigilancia en las costas.

3 Arquitectura del sistema

3.1 Idea final del proyecto.

La pretensión final de este trabajo, como viene adelantándose en los epígrafes previos, es diseñar y desarrollar un sistema capaz de medir cambios o movimientos bruscos de la postura corporal a partir de los datos obtenidos por tres sensores inerciales. Para convertir este objetivo en una realidad, se ha partido de un primer proyecto, que consiste en el análisis, filtrado y postprocesado de los datos obtenidos de un sensor, para luego poder aplicar los conocimientos adquiridos al desarrollo final.

No cabe de duda, por tanto, de que el proceso de diseño parte de la elección tanto del microcontrolador como del sensor que utilizaremos en nuestro estudio. Al tratarse finalmente de un proyecto que formará parte de un proyecto más amplio, uno de los criterios a la hora de seleccionar nuestros dispositivos se centró en su sencillez en cuanto a la programación y conexión de los mismos. De hecho, el primer ejercicio realizado se centró en el estudio de diferentes dispositivos comercializados, atendiendo a sus características técnicas diferenciales y a su ratio calidad precio.

Abordamos en primer lugar la selección del microcontrolador más adecuado y, siguiendo los criterios mencionados anteriormente, la elección se decantó por el AtMega32U4 desarrollado por Atmel. Este microcontrolador está presente en varios de los proyectos desarrollados en el grupo de investigación donde se ha desarrollado este TFG y consecuentemente su manipulación y programación no presentaría, a priori, una gran dificultad. Adicionalmente, fue considerado un atributo especialmente atractivo el hecho de que este microcontrolador formase parte de varias placas de desarrollo Arduino, como Leonardo y Mega, un rasgo que nos facilitaría enormemente el proceso de obtención de los datos en una primera fase de nuestro desarrollo.

Tomada esta decisión, nos enfrentamos a la elección de la IMU más adecuada, cuya función no será otra que la de obtener los datos que se precisan para la creación de nuestro prototipo. En el mercado existen cientos de estos sensores, con diferentes encapsulados, grados de libertad e incluso con diferentes sensores en su interior. Es por ello que en el proceso de decisión se escrutaron un amplio número de posibilidades, realizando un pequeño estudio con algunas de las IMU más interesantes que encontramos en las principales tiendas de componentes electrónicos especializadas de las que teníamos referencias.

Dicho estudio, reflejado en la Tabla 3.1, nos condujo a la incorporación de criterios adicionales en el proceso de decisión, cuyo resultado fue la elección de la MPU-9250 de Sparkfun como la más acertada, al presentar ventajas relativas con relación al resto de sensores analizados.

Para tomar nuestra decisión final en cuánto a que sensor utilizar en nuestro estudio, se tuvieron en cuenta distintos factores. Por un lado, se decidió realizar el estudio sobre una IMU que contara con los máximos grados de libertad posibles, por lo que se redujeron las opciones barajadas en un primer momento a dos, la ICM-20948 y la MPU-9250. Por otro lado, a igualdad de grados de libertad, se decidió un encapsulado SIP, caso 6 en la tabla. Además, un factor diferencial en la elección fue la existencia a un precio razonable del sensor ya soldado a una placa lista para desarrollar sobre ella.

Adicionalmente, la posibilidad de tener diversas librerías ya hechas por el propio fabricante a nuestra disposición, así como la documentación aportada por el mismo, han resultado determinantes a la hora de seleccionar este sensor, que se presentaba como una alternativa accesible y con garantías de éxito.

Tabla 3 Criterios aplicados para la selección del sensor.

<i>Nombre Link del proveedor</i>	<i>Encapsulado</i>	<i>Dimensiones [mm]</i>	<i>Precio (€)</i>	<i>Chips / libertad</i>
ICM-20948 http://www.components-center.com/product/InvenSense/ICM-20948.html	24QFN	3x3x1	2,83	A,G,M / 9
FIS 1100 http://www.onsemi.com/PowerSolutions/product.do?id=FIS1100	LGA 16	3,3x3,3x1	8,10	A,G / 6
BMI 055 https://www.digikey.com/product-detail/en/bosch-sensortec/BMI055/828-1043-1-ND/4196679	LGA	3x4,5x0,95	6,38	A,G / 6
BMI 160 [136-0828] http://es.rs-online.com/	LGA	2,5x3x0,8	24,96	A,G /6
MPU 6050[883-7948] http://es.rs-online.com	QFN	4x4x0,9	8,80	A,G /6
MPU 9250 https://www.sparkfun.com/products/13762	SIP	3x3x1	13,22	A,G,M /9

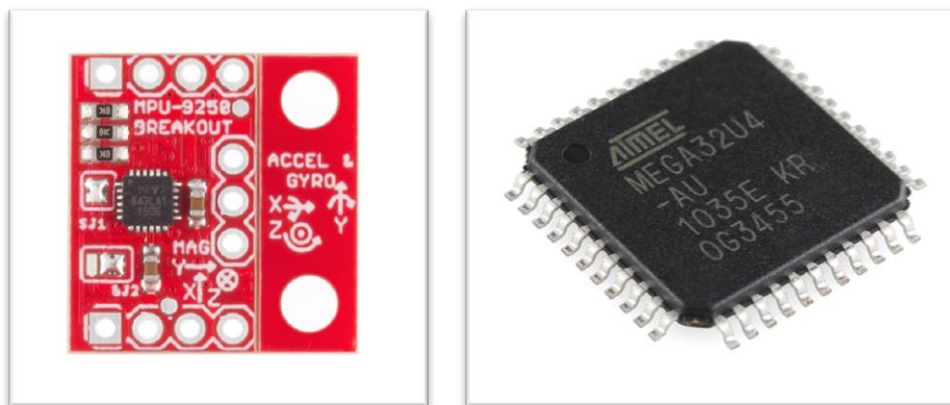


Figura 1 Dispositivos seleccionados: Sensor MPU-9250 incorporado a una PCB y Microcontrolador AT Mega32U4.

Una vez que fueron seleccionados los componentes que formarían el núcleo del proyecto, se planteó el esquema de desarrollo del mismo basado en tres etapas.

En la primera se llevó a cabo el diseño utilizando el sensor y una placa de desarrollo Arduino, para desarrollar el software de adquisición y representación en un entorno conocido y con una amplia bibliografía a nuestra disposición. Posteriormente y para reducir el tamaño del diseño, se desarrolló la misma arquitectura eliminando Arduino, sustituyéndolo por el ATmega32U4, ahora integrado en una PCB comercial, adecuando el software desarrollado en la etapa anterior a la nueva plataforma. En la tercera etapa se diseñó una PCB para integrar todos los elementos, considerándose este como el diseño final.

La exposición gráfica de este proceso se recoge en el diagrama de flujos que aparece a continuación:

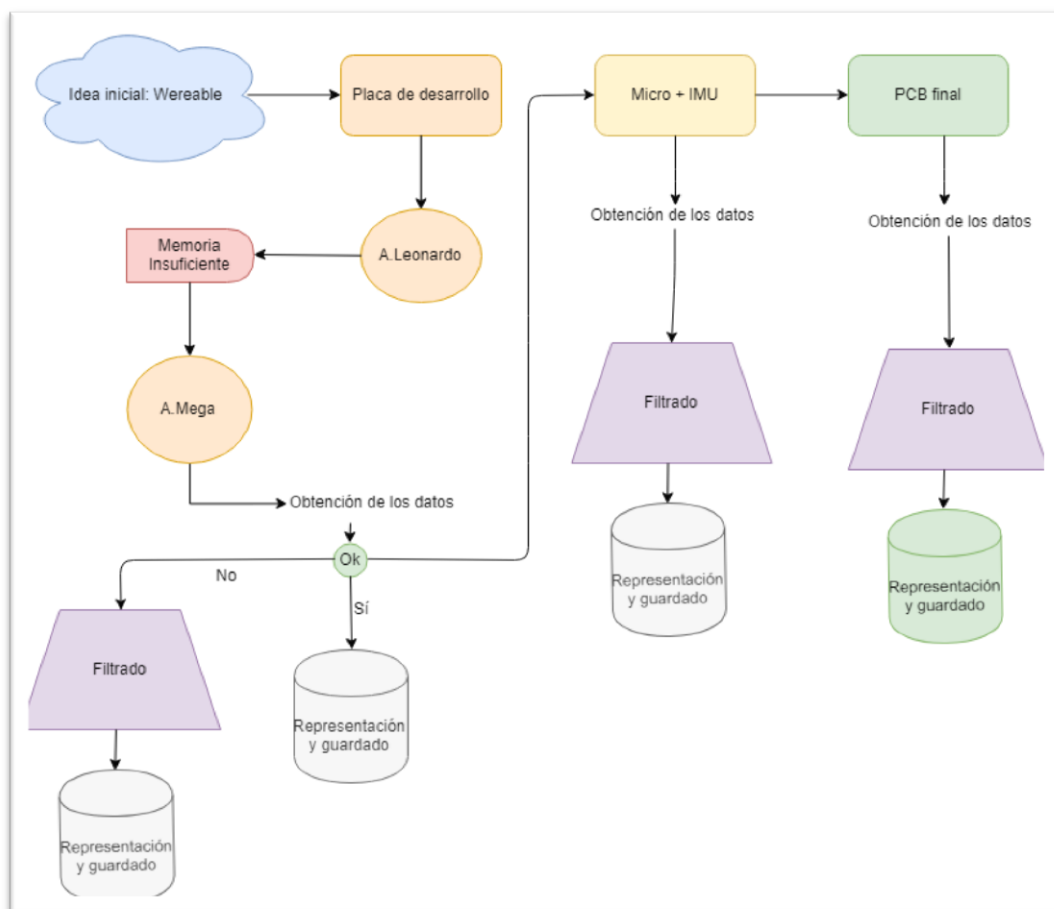


Figura 2 Esquema del flujo de diseño.

3.2 Diseño en placa de desarrollo Arduino

Como se describe en el diagrama de flujos previo, para esta primera parte del desarrollo utilizamos una placa de desarrollo Arduino Leonardo, que contiene el AtMega32u4, es decir, sensor y microcontrolador seleccionados. Este primer diseño cubre el objetivo planteado en términos de dominio de conocimiento, es decir, se ha creado con el propósito de ayudarnos a obtener y comprender los datos del sensor, puesto que hay numerosas librerías en Arduino. Este desarrollo nos facilitó, en gran medida y tal y como se había planteado, un importante nivel de conocimiento interno sobre el comportamiento del sensor, siendo éste de gran ayuda para ejecutar el resto del diseño.

Por su parte, el sensor MPU9250, cuenta con los tres sensores anteriormente mencionados, acelerómetro, giróscopo y magnetómetro y todos ellos cuentan con tres ejes, lo que otorga al dispositivo de nueve grados de libertad. Este sensor se utiliza en la actualidad en diversas aplicaciones de navegación y geolocalización, así como en aplicaciones aeroespaciales [12] [16], que han puesto de manifiesto su versatilidad y eficiencia.

Para la conexión de ambas partes se utilizarán los pines SDA y SCL de ambas placas, que pueden ser localizados con facilidad en los esquemáticos de ambos.

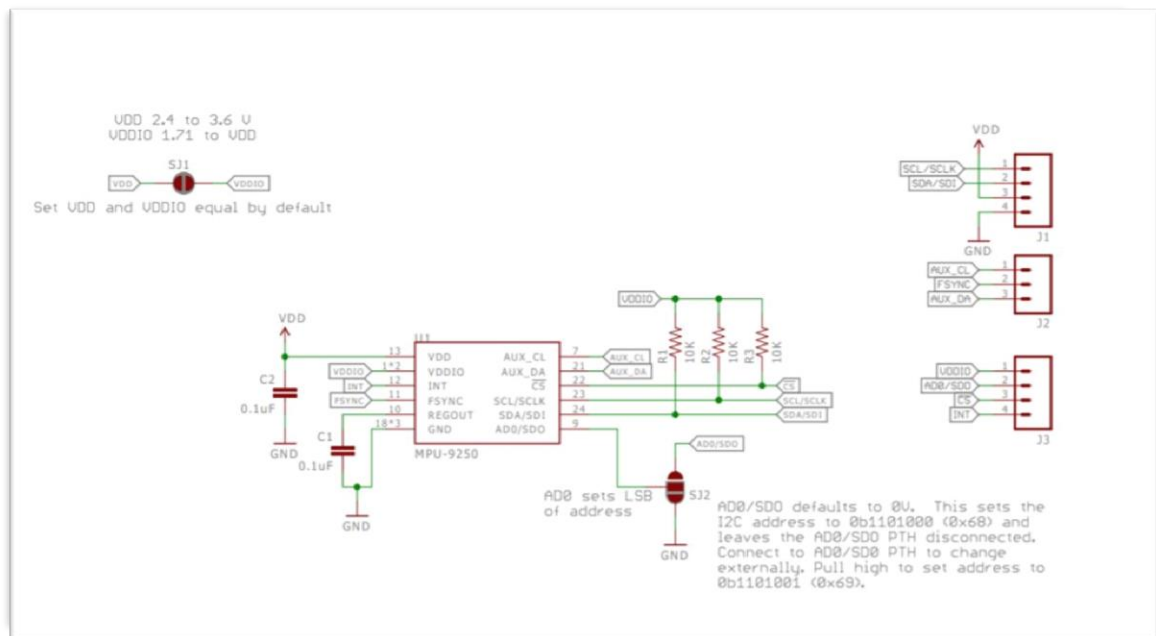


Figura 3.3- Esquema del flujo de diseño

Tabla 4 Conexiones entre la IMU y la placa de desarrollo mediante I2C.

MPU9250	Arduino Mega
SDA	Pin 20 (SDA)
SCL	Pin 21(SCL)
VCC	3,3 V
GND	GND

Tabla 5 Direccionamiento del I2C

AD0(nivel lógico)	Dirección I2C
“0”	0x68
“1”	0x69

En nuestro caso, el valor lógico del AD0 será “0” y por tanto nuestra dirección del I2C 0x69.

Una vez disponemos de dicha conexión para poder observar los datos medidos, se procedió a su transmisión, mediante el puerto serial, al programa de análisis deseado. En nuestro caso para este primer ensayo utilizamos Matlab, herramienta de desarrollo que nos permitirá la transmisión y representación de los datos de una manera sencilla.

El núcleo de la aplicación ha sido desarrollado en lenguaje C++ y la parte más superficial en lenguaje propio del IDE de Arduino [Anexo-C].

La conexión física de ambos dispositivos se muestra en la figura siguiente.

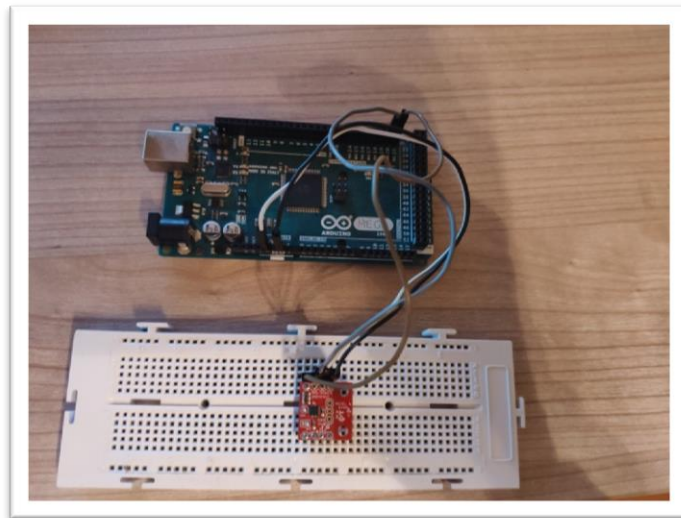


Figura 3 Conexión física de los dispositivos.

3.3 Diseño en PCB final

Una vez entendido el funcionamiento del sensor, y analizados los datos obtenidos en este primer diseño con la placa de desarrollo, se pasó a su desarrollo sobre PCB, utilizando el microcontrolador independiente de la placa de Arduino.

Para realizar este diseño se programó el microcontrolador en C para posteriormente y utilizando una protoboard para realizar las conexiones en un primer momento, conectar el microcontrolador a la IMU.

En nuestro caso y por simplicidad elegimos una placa de Sparkfun, llamada pro micro, en la que se incorpora el procesador seleccionado y, además, permite reutilizar el código Arduino desarrollado previamente para programarlo en lugar de programarlo íntegramente en C.

Las especificaciones técnicas en cuanto a voltajes y corrientes de la tarjeta Pro Micro se encuentran resumidas en la siguiente tabla:

Tabla 6 Especificaciones del chip Sparkfun pro-micro.

Especificación	Valor []
Voltaje entrada (USB)	5 [V]
Voltaje de salida (Vcc)	3,3 [V]
Corriente máxima chip	200 [mA]
Corriente recomendada por pin	20 [mA]
Frecuencia máxima del micro	16 [MHz]
Memoria flash	32 [kB]

cabo una primera aproximación utilizando una pila para darle la alimentación necesaria a la placa, anteriormente proporcionada mediante la conexión USB.

Se buscaron diferentes modelos de pilas de botón, teniendo en cuenta los requisitos de corriente y voltaje necesarios para el correcto funcionamiento del circuito. Estos requerimientos hacían que nuestra pila precisase incorporar dos características principales: de un lado debería dotar de una tensión continua de 3,3 a la placa del microcontrolador, y por otro lado suministrar la suficiente corriente tanto al sensor como a la placa pro-Micro.

El primer requisito sería fácilmente abordable utilizando un regulador de tensión a la tensión deseada, por el contrario, el segundo requerimiento ya no se nos presentaba tan abordable como el anterior. La Pro-Micro consume una gran cantidad de corriente y no encontramos una pila de un tamaño y precio ajustados para proporcionar dicha corriente.

Es en este momento fue cuando se planteó una segunda opción, utilizar una batería portátil, “power bank” la cual será la encargada de suministrar tanto el voltaje como la corriente necesaria al wearable. El uso de este tipo de batería, además de atender nuestras exigencias, contaba como gran ventaja su facilidad de recarga con un simple cargador universal micro USB.

Sin embargo, este planteamiento, por el momento no ha podido ser aplicado y se plantea como una posible mejora futura del proyecto. Su inviabilidad reside en que la carga del wearable mediante estas baterías, no puede ser aplicada ya que estamos realizando la transmisión de los datos mediante el puerto serie USB. Por tanto, para poder alimentar el circuito de la forma mencionada anteriormente, debemos incorporar un tercer elemento a la PCB, un módulo para la transmisión de los datos de forma inalámbrica, ya sea un módulo Bluetooth o un módulo Wifi.

Una vez estudiadas las alternativas para suministrar energía al circuito, y tras continuar alimentándola desde el ordenador, llegó el momento de trasladar nuestro esquemático al software para la fabricación de nuestra placa.

Para la realización de este diseño utilizamos los esquemáticos facilitados por el fabricante de los dispositivos, en este caso Sparkfun, para poder, a partir de ellos, generar los componentes en el software Altium Designer, e integrarlos en nuestra PCB.

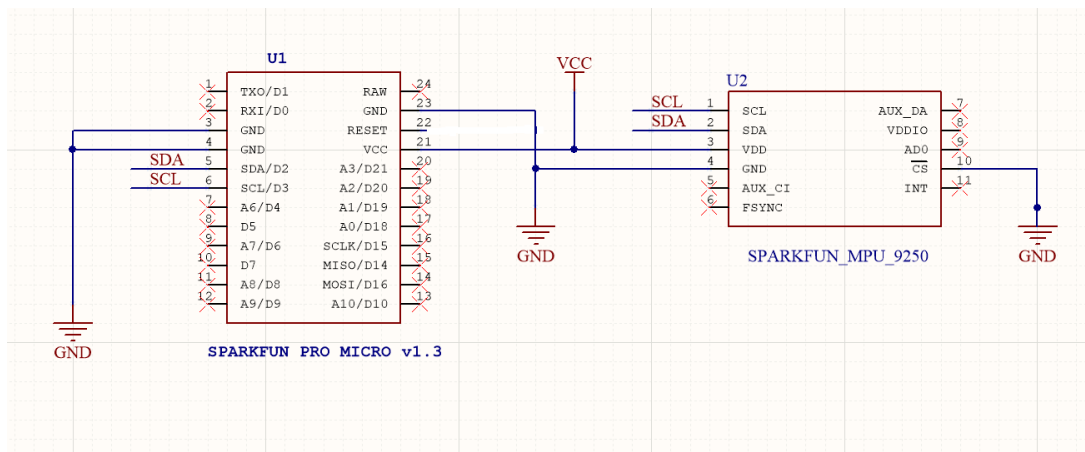


Figura 6 Esquemático de la placa proMicro de Arduino extraída de [17].

En este proceso fue preciso abordar, en primer lugar, la creación del esquemático para poder visualizar las conexiones que se necesitaban. Una vez creado dicho esquemático, nos dispusimos a realizar las huellas, “footprints” con los pines necesarios para el enrutamiento de nuestras pistas. Por último, se generaron los archivos gerber y de taladros necesarios para mandar a fabricar nuestra PCB.

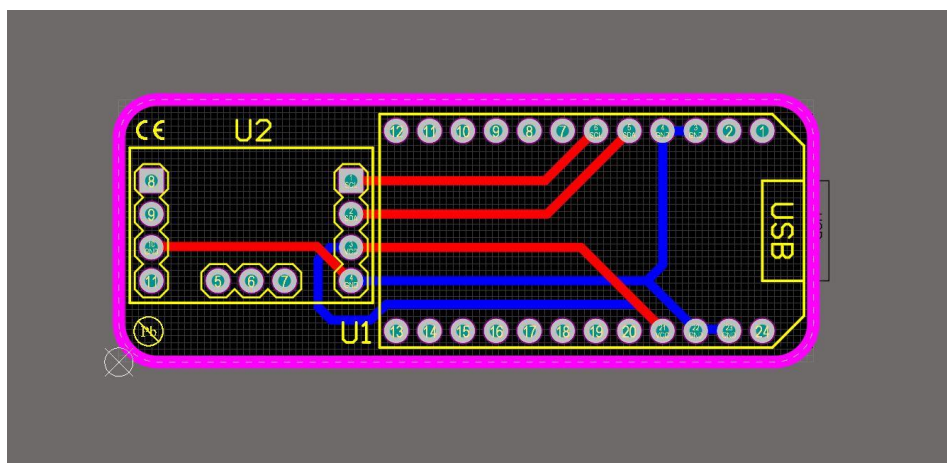


Figura 7 Esquemático del enrutamiento de la placa

Una vez fabricada la PCB, se procedió a la comprobación de que todas las pistas se habían generado correctamente, procediendo a su soldado, tanto de la IMU como el microcontrolador a ella, quedando como se observa en las siguientes figuras:

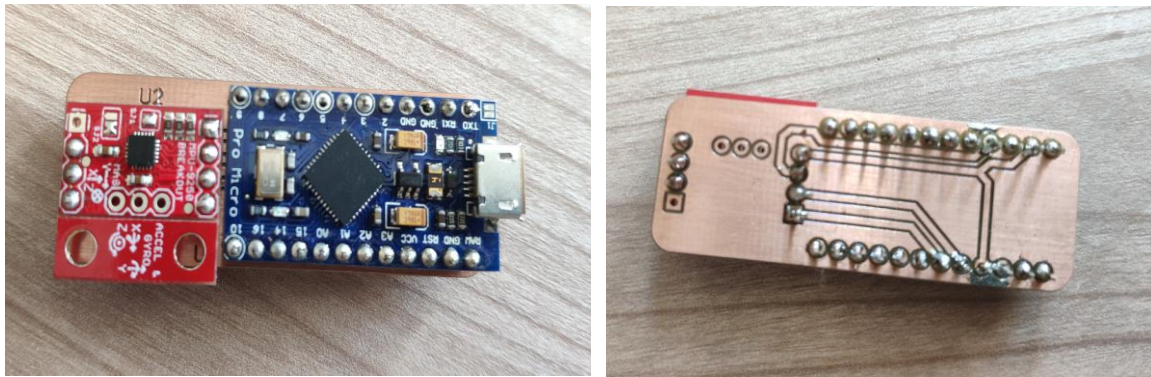


Figura 8 ATmega32U4 y MPU 9250 soldados en la nueva PCB

Para terminar nuestro prototipo, decidimos fabricar una carcasa de cara a una posible presentación de un producto final. Al tratarse de un primer paso en el total de un proyecto wearable, esta carcasa no sería la definitiva del producto, pero si nos da una idea de cómo se podría presentar un producto final.

En el producto final sobrarán muchos de los elementos incluidos en las dos placas PCBs que contienen el sensor y el procesador. Por otro lado, hay que tener en cuenta que el producto final debe incluir como mínimo, un chip para la comunicación inalámbrica de los datos sensados y procesados, así como un regulador de tensión para el sistema de alimentación comentado previamente. Estas evidentes mejoras de cara a obtener un producto final quedan fuera del alcance de este trabajo y serán propuestas como parte del trabajo futuro de continuación de este TFG.

Para la realización de esta carcasa, se ha utilizado una impresora 3D facilitada por el Club de Robótica de la UAM, una vez definido el diseño del soporte que se ajustaba a nuestras necesidades. En concreto se ha optado por una caja rectangular con un cierre, escalonando sus medidas del dispositivo wearable, cuyo resultado final se muestra en la figura siguiente:



Figura 9 Posible idea de presentación final

4 Desarrollo

4.1 Desarrollo en Arduino

4.1.1 Análisis previo, datos en raw o DMP.

Como se ha visto en el apartado de diseño, nuestro proyecto se desarrolla en 3 etapas. En la primera de ellas nos enfocamos en el estudio y aprendizaje de los distintos componentes que contendrá el proyecto final utilizando una placa de desarrollo, en este caso elegimos Arduino como plataforma.

Dentro del universo Arduino elegimos en un primer momento la placa Leonardo. La elección de este modelo en concreto se basó en dos puntos principales. Por un lado, el microcontrolador que alberga esta placa es un microcontrolador que conocíamos en el grupo y, por otro, el bajo coste de la misma aconsejaba su uso puesto que se trataba de una placa que tan solo se utilizaría en una primera fase del desarrollo, dos razones de peso que hicieron decantarnos por ella.

Una vez elegidas tanto la IMU a utilizar como la placa de desarrollo, era el momento de enfocar todos nuestros esfuerzos en la obtención de los datos del sensor, para llevar a cabo nuestro proyecto.

Tras realizar un estudio en profundidad del chip en concreto, fuimos conscientes de que se podría disponer de dos métodos para obtener los datos del mismo. Directamente del sensor, o a través del DMP integrado en la placa. El DMP (Digital Motion Processor), es un módulo integrado que nos permitiría obtener ya filtrados y tratados los datos que buscamos.

Es decir, los datos que podemos obtener directamente de la IMU son, los que obtenemos de cada uno de los sensores que la integran. Dichos datos, sin un tratamiento previo (en crudo) son muy difíciles de manejar, ya que se trata de aceleraciones de la gravedad, velocidades angulares y fuerzas de campo magnético, que precisan de un proceso de filtrado y de tratamiento previo para poder observar la variación del movimiento, nuestro objetivo final.

Considerando esta circunstancia, se decidió utilizar el DMP, puesto que nos posibilita la opción de disponer directamente de los ángulos de Euler o los cuaterniones, mediante los cuales podemos observar, y medir las variaciones en el movimiento, añadiendo a esta ventaja la de poder contar con el código proporcionado por el propio fabricante.

En este código pudimos observar dos partes diferenciadas para poder obtener los datos deseados. En una primera parte debíamos inicializar el DMP, otorgándoles unos valores predefinidos a unas constantes. Tras esta primera fase, se llamaría a la función que actualizará la cola FIFO con los datos obtenidos por el sensor, obteniéndose en este proceso tanto los ángulos de Euler como los cuaterniones asociados. Sin embargo fue en este punto en el que nos enfrentamos con el primer gran inconveniente, el modelo de placa de desarrollo seleccionada no contenía la suficiente memoria interna como para albergar todo el código facilitado por el fabricante para la utilización del DMP.

Nos decantamos entonces por simplificar dicho código, utilizando únicamente las funciones de activación y calibración del DMP junto con el método para obtener los cuaterniones, pero ni con la reducción del código conseguimos tener la suficiente memoria como para cargarlo en la placa.

Fue en este momento cuando decidimos cambiar el modelo de la placa de desarrollo. Optamos por una placa con el mismo modelo de microcontrolador, para poder mantener nuestro código lo más parecido posible, pero que contuviera una memoria mayor, en este caso la Arduino Mega.

Con esta nueva placa de desarrollo pudimos cargar sin problema el código proporcionado por Sparkfun para la utilización del DMP integrado en la placa. Pese a poder utilizar íntegramente el código del fabricante, sin ninguna simplificación, y tras numerosas pruebas en las que seguimos sin obtener resultados aclaradores, se llegó a la conclusión de que el código expuesto por el fabricante contaba con algunas imprecisiones que hacían inviable su uso directo.

Ante esta situación, se inició una investigación profunda sobre el origen de nuestras dificultades y, acudiendo a distintos foros encontramos que efectivamente dicho código contenía errores, y debíamos cambiarlo sustancialmente para poder ser utilizado. De hecho, ante este escenario se decidió en este momento no utilizar el DMP y utilizar directamente los datos que nos proporcionaban los sensores para calcular el movimiento en tiempo real detectado por el sensor.

Se barajaron dos posibles alternativas para, a partir de los datos de la IMU, obtener el movimiento que buscamos. Por un lado, podíamos calcular los ángulos de Euler, α , β y γ y, por otro, se consideró la opción de trabajar con cuaterniones, que a priori nos ofrecen algunas ventajas respecto a los ángulos anteriormente mencionados.

La elección de unos u otros, las ventajas y desventajas de cada uno, y la obtención y utilización de los mismos a partir de los datos obtenidos del sensor pasó a ser parte fundamental de este trabajo. Por ello trataremos este estudio en el anexo A de esta memoria.

No obstante, los resultados de este análisis, nos ha llevado a considerar que lo más conveniente sería utilizar los ángulos de Euler, más concretamente una variante de estos utilizada en navegación, los ángulos roll pitch y yaw, ya que además de resultar claramente más intuitivos en términos de interpretación e incluso a la hora de graficarlos, los problemas e inconvenientes que iban surgiendo en el proceso de su aplicación real, frente a los que surgían con el uso de los cuaterniones, resultaron más abordables, al menos para nuestro caso de estudio.

4.1.2 Obtención de los ángulos de Euler

En un primer momento se calcularon los ángulos de inclinación del sensor de una manera matemática sencilla a partir de las aceleraciones obtenidas directamente de la IMU como se muestra en la siguiente figura.

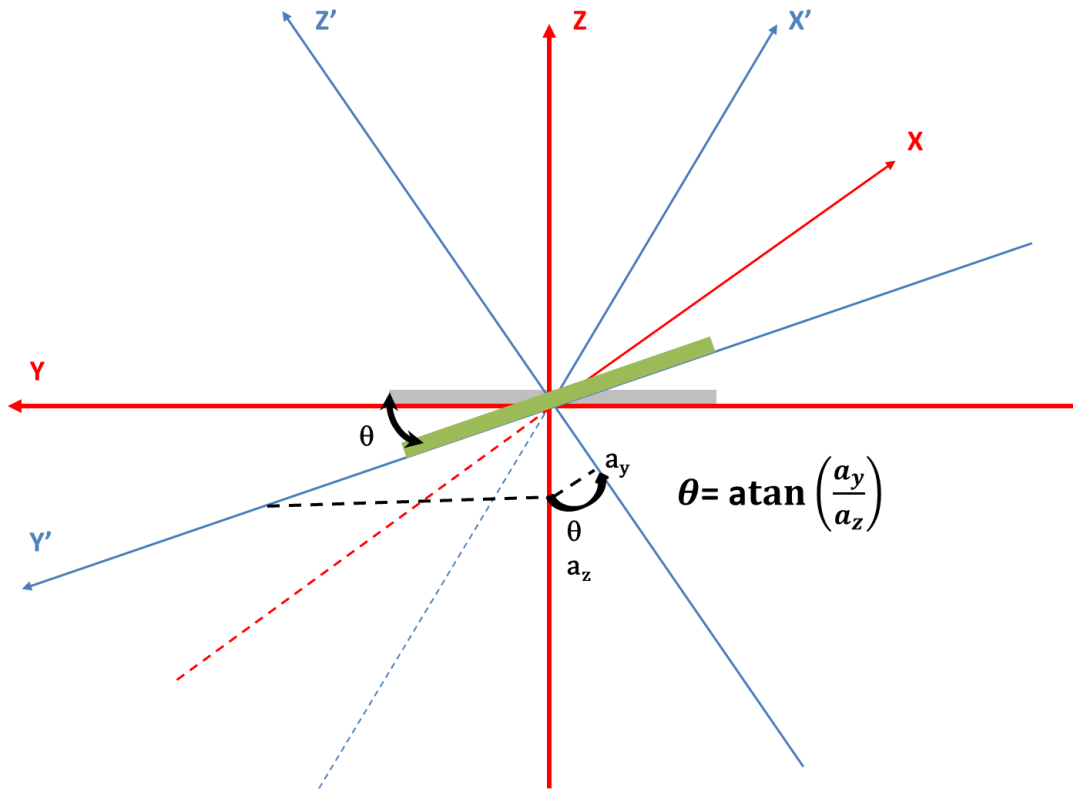


Figura 10 Aproximación al cálculo de los ángulos de rotación mediante aceleraciones.

Para la obtención de los datos provistos por el sensor, fue preciso apoyarnos en tres librerías principales.

- La primera de ellas será la librería Wire.h de Arduino, que contiene las funciones necesarias para la comunicación serie con el dispositivo.
- Las otras dos librerías son IMU_MPU9250.h, una librería propia del fabricante del sensor en la que encontramos las funciones para la inicialización, el calibrado del sensor así como para el tratamiento de los valores que se obtienen del acelerómetro y el giróscopo. MadgwickAHRS.h es la segunda librería que contiene el filtro Madgwick que utilizaremos para en primer lugar integrar los datos obtenidos por los sensores, y posteriormente para filtrar el ruido de las señales que nos interesan y compensar la deriva que se acumula con el tiempo en los datos obtenidos, concretamente del giróscopo. Se ampliarán detalles sobre la elección y funcionamiento de este filtro de Madgwick en el anexo B de esta memoria.

En una primera aproximación tan solo utilizamos la señal en crudo obtenida del acelerómetro, una señal que además de contener demasiado ruido hizo que los cálculos de los ángulos no resultaron tan precisos como se requería. A esta limitación se le añade el

hecho de que solo podría realizarse considerando una de las dimensiones y, por tanto, no podríamos obtener la orientación espacial.

Por todo ello se pasó a una segunda aproximación en la que se utilizaron los datos obtenidos de ambos sensores junto con un filtrado de la señal. Este proceso se iniciaba con la llamada a las funciones que calibran e inicializan el sensor. Posteriormente se incorporan los datos proporcionados por el acelerómetro y el giróscopo. En particular, para nuestro caso de estudio, estos datos serán los utilizados para obtener la precisión deseada en nuestras medidas de los ángulos de Euler y no será necesaria la utilización del magnetómetro.

Cerrada esta etapa se inicializa el filtro con una frecuencia de 40 KHz, o lo que es lo mismo 25 microsegundos. Esta frecuencia es fundamental, ya que si el filtrado no se realiza de forma suficientemente rápida se acumula mucho error a la hora de calcular los 3 ángulos de Euler. Inicializado el filtro, con la frecuencia que más se ajusta a nuestras necesidades, de la misma forma que se ha procedido a inicializar y calibrar el sensor inercial, estábamos en disposición de obtener los datos del acelerómetro y el giróscopo, a través de la lectura de los 6 registros en raw de ambos elementos, para finalmente obtener los datos de las aceleraciones multiplicando los datos obtenidos por un factor de escala.

Precisamente la medición de las aceleraciones, correctamente calculadas, han de ser filtradas para eliminar los posibles errores de ruido y deriva, utilizándose para ello el filtro de Madgwick, de cuyo proceso se obtienen los cuaterniones correspondientes a las aceleraciones obtenidas del sensor y, a partir de estos, se pasa a definir cuantitativamente los ángulos de Euler, que serán utilizados para la representación en tiempo real de nuestro movimiento.

4.1.3 Representación del movimiento en tiempo real

Llegados a este punto, ya habíamos obtenido los datos del acelerómetro y el giróscopo, con su tratamiento habíamos conseguido disminuir notablemente tanto los errores propios de los sensores (deriva, desviación) como los introducidos en el proceso de obtención de los datos (ruido) y se había conseguido obtener los cuaterniones y, a partir de estos, los ángulos de Euler. En definitiva, estábamos en condiciones de cubrir la etapa de obtener la representación del movimiento en tiempo real.

Para ello, se definió que la forma con la que podríamos conseguir representar el movimiento a través de los ángulos de Euler no podría ser otro que el uso de las matrices de rotación.

Comenzamos, por tanto, definiendo qué es la matriz de rotación y cómo podemos utilizarla para obtener un movimiento en 3 dimensiones. Para nuestro caso de estudio, se precisa de tres matrices definidas en R^3 . Mediante dichas matrices se puede obtener la rotación de un vector alrededor de cada uno de los 3 ejes principales x, y, z, quedando definidas de la forma siguiente:

- Matriz de rotación alrededor del eje x:

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix}$$

- Matriz de rotación alrededor del eje y:

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

- Matriz de rotación alrededor del eje z:

$$R_z(\phi) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Es decir, dado un vector v_1 en el espacio tridimensional R^3 , una rotación de dicho vector v_1 alrededor del eje “x” con un ángulo ψ , vendría dado por la expresión:

$$V'_1 = R_x \times V_1$$

De igual manera la rotación de ese mismo vector alrededor del eje “y” un ángulo θ vendría dado por la expresión:

$$V'_1 = R_y \times V_1$$

Y podríamos repetir el mismo proceso para una rotación alrededor del eje z con la matriz R_z .

Pero, ¿qué ocurre cuando queremos realizar una rotación a lo largo de varios ejes?

Bien podríamos entender esta rotación como el producto de matrices de rotación, tantas como ejes en los que queramos computar o valorar esta rotación. Obteniendo de este modo la matriz genérica de rotación R , que vendría definida por la expresión:

$$R = R_x(\phi)R_y(\theta)R_z(\phi)$$

Ahora bien, el producto de matrices no es conmutable, por tanto, en la ecuación anterior estaremos definiendo un movimiento de rotación en los tres ejes de manera secuencial, primero en “x” luego en “y” y por último en “z”.

Para nuestro caso de estudio no será de vital importancia el orden en el que realicemos las operaciones, y por tanto la manera en la que obtengamos la matriz de rotación para definir el movimiento.

De hecho, en la siguiente ecuación se muestra uno de los múltiples ejemplos de matriz de rotación para la composición de los tres giros, aunque existen otras según el orden en que realicemos la multiplicación de las matrices de rotación “elementales”

$$R = R_x(\varnothing)R_y(\theta)R_z(\varphi)$$

$$= \begin{pmatrix} \cos \theta \cos \varnothing & \sin \varphi \sin \theta \cos \varnothing - \cos \varphi \sin \varnothing & \cos \varphi \sin \theta \cos \varnothing + \sin \varphi \sin \varnothing \\ \cos \theta \sin \varnothing & \sin \varphi \sin \theta \sin \varnothing + \cos \varphi \cos \varnothing & \cos \varphi \sin \theta \sin \varnothing - \sin \varphi \cos \varnothing \\ -\sin \theta & \sin \varphi \cos \theta & \cos \varphi \cos \theta \end{pmatrix}$$

Ecuación 4.1- Matriz de rotación.

Para la representación del movimiento en 3 dimensiones en tiempo real, se utilizará el programa Processing. Se ha elegido este programa ya que nos proporciona una manera sencilla de dibujar cuerpos en el espacio tridimensional y moverlos a partir de la matriz de rotación calculada previamente.

Para nuestro modelo se ha decidido mostrar la figura de nuestra IMU como ejemplo, pero se podrían hacer otras figuras geométricas e incluso modelar la figura de una persona para la representación de dicho movimiento.

En esta fase de nuestro desarrollo surgen numerosos problemas debido a la complejidad que supone la obtención de los datos del sensor mediante software. Algunos de los más relevantes se detallarán a continuación, considerando que han significado una ampliación del método de aprendizaje que, previsiblemente, facilitará desarrollos futuros.

En un primer lugar, y ante la posibilidad de que la obtención de los datos mediante software fuera demasiado laboriosa, se decidió utilizar sin éxito el DMP integrado en el sensor, debido a que el código proporcionado por el fabricante contenía errores y no se consiguieron valoraciones efectivas y válidas de las variables que se precisaban.

Una vez centrados en la obtención de los datos mediante software nos encontramos con otro de los grandes problemas de este trabajo, la transformación de los datos. Nos surgen grandes problemas para obtener datos representables en tiempo real para la definición del movimiento a partir de los datos que nos facilita el sensor. El tratamiento de estos datos para obtener el movimiento está tratado a lo largo de esta memoria con detalle, por lo que no nos detendremos en ello, evitando duplicidades, nuevamente en esta sección.

Como ejemplo de los resultados obtenidos en esta sección, se muestran en la siguiente figura los ángulos de navegación obtenidos en ella mediante Matlab.

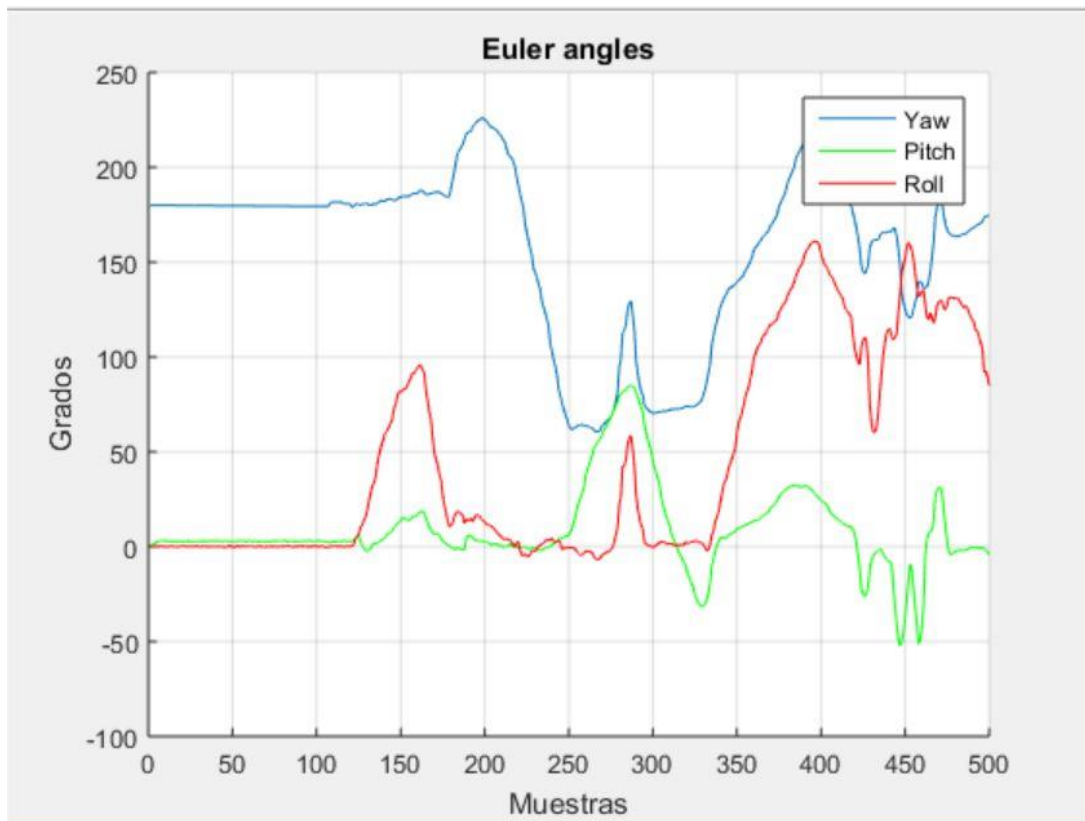


Figura 11 Representación de los ángulos de navegación.

Como se puede observar en la figura, se tomaron 500 muestras realizando un movimiento arbitrario del sensor y se graficaron por separado los tres ángulos de navegación, rol, pitch y yaw medidos en grados (°).

4.2 Desarrollo con microcontrolador en protoboard

El objetivo se centraba en eliminar la dependencia con Arduino, ya que su tamaño se presentaba excesivo como para hacer posible un desarrollo a modo de wearable.

Para ello adquirimos el microcontrolador Atmega32u4 integrado en una pequeña placa, la Sparkfun Pro-Micro a 16 MHz. Esta placa nos permitió reducir significativamente el tamaño de nuestro dispositivo, pero a la vez reutilizar nuestro código Arduino para programar el microcontrolador.

Tras realizar algunos ajustes en el entorno Arduino IDE, se procedió a programar el nuevo micro para posteriormente comprobar que los valores obtenidos eran los mismos que los encontrados en la etapa anterior.

4.2.1 Subsección Problemas y aprendizaje

Como hemos mencionado, debido a los problemas surgidos anteriormente a la hora de obtener los datos del sensor, los plazos para el desarrollo del trabajo se habían resentido y éramos conscientes de lo crucial que resultaría realizar una buena elección del microcontrolador más adecuado.

De nuevo una inmersión específica y profunda de la oferta existente nos llevó a la decisión de utilizar la Sparkfun Pro-Micro debido a, que por un lado, que sus especificaciones técnicas en cuanto a consumos de corriente no eran excesivamente altas, y por otro, nos permitía utilizar casi en su integridad el código Arduino ya programado, lo que nos permitía reducir significativamente los tiempos de desarrollo.

Sin embargo, para poder integrar el código Arduino en dicha placa se tuvo que añadir al entorno de desarrollo Arduino IDE nuestra placa, utilizando para ello el tutorial presente en la página del fabricante [25], en el que se detalla paso por paso como conseguir que el entorno reconozca la nueva placa.

En esta fase de nuestro desarrollo no surgieron problemas significativos debido a que en la fase anterior se depuró bastante el código de manera que esta fase supusiera una mera transición entre el modelo con Arduino y el modelo final en PCB.

4.3 PCB final

La última fase del proyecto se centró en la integración de ambos dispositivos, el microcontrolador y el sensor IMU mediante una PCB que previamente se tuvo que diseñar y fabricar.

En principio el diseño de esta placa debería ser relativamente sencillo puesto que disponíamos de un sustrato bicapa, es decir, de ambas caras de la PCB para enrutar nuestras pistas, que además no eran demasiadas.

Para el diseño de nuestra placa utilizamos el software de Altium, Altium Designer en su versión disponible en los laboratorios de la EPS. Junto a este, fue preciso llevar a cabo el diseño de nuestro esquemático, iniciándose este procedimiento con la búsqueda de las huellas (layout) de nuestros dispositivos en librerías proporcionadas por los fabricantes de las mismas. Al tratarse de dispositivos ampliamente utilizados en el mercado, la búsqueda de estas huellas nos resultó relativamente sencilla y abrieron paso a la realización del diseño esquemático entre el microcontrolador y el sensor.

Para ello nos basamos en las conexiones físicas que teníamos en nuestra fase anterior, de modo que tan solo tuvimos que conectar las salidas de datos, el reloj, la alimentación y la tierra, un proceso del que partió el enrutamiento de las pistas.

Como hemos mencionado anteriormente, contábamos con un sustrato bicapa, por lo que esta tarea al contar con un número de pistas limitado y ambas caras para poder realizarlo no supuso un gran esfuerzo, y se pudo realizar sin mayores dificultades.

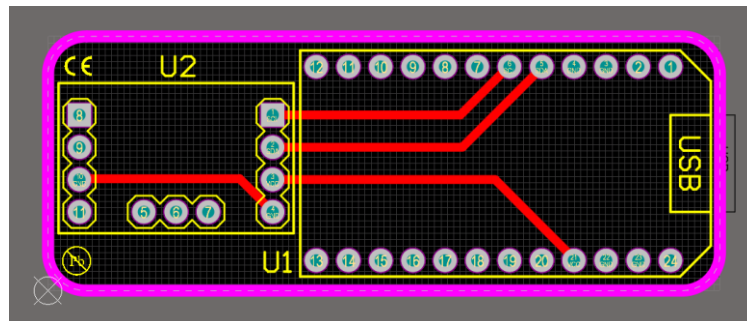


Figura 12 Enrutamiento de la capa superior

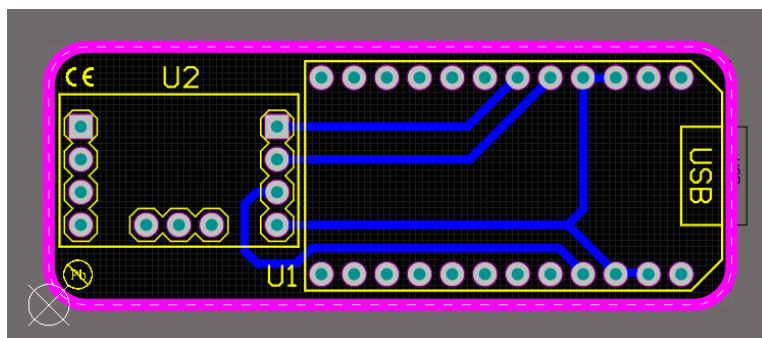


Figura 13 Enrutamiento de la capa inferior.

Finalmente, para poder solicitar la fabricación de nuestra PCB fue preciso generar los ficheros gerber y el fichero de taladros, requisitos exigidos por el taller para la correcta fabricación de nuestra placa.

Una vez fabricada la placa, el siguiente paso fue soldar ambos componentes y comprobar mediante un multímetro digital, que las soldaduras estaban correctamente realizadas y que los contactos estaban como debían estar, de tal forma que efectivamente, podíamos representar al igual que hacíamos en las etapas anteriores, el movimiento en tiempo real del sensor.

Quedaba, por tanto, como última fase conectar nuestra placa al ordenador, cargar nuevamente el programa y comprobar que habíamos sido capaces de representar el movimiento de la misma nuevamente mediante el software Processing. Una etapa que fue resuelta con éxito. Como evidencia de ello, durante la presentación se mostrará un video a modo de demostración de los resultados obtenidos, en el cual se podrá observar el movimiento del sensor representado en Processing en tiempo real.

4.3.1 Subsección Problemas y aprendizaje

Esta última fase del desarrollo presenta como rasgo fundamental su interconexión con desarrollos fundamentalmente de hardware, puesto que se parte de unos resultados obtenidos mediante software que ya están testados. Es por ello que todos los problemas que nos han surgido, y todo lo aprendido durante la realización de esta fase se centran en el diseño y fabricación del hardware.

Durante este desarrollo, se ha aprendido a utilizar más en profundidad la herramienta de desarrollo de Altium para la fabricación del PCB, donde encontrar los diferentes menús, como realizar un enrutamiento bicapa en una superficie pequeña, como pasar de nuestra idea al esquemático final a partir de las huellas, a buscar y transformar las huellas facilitadas por el fabricante al formato que necesitamos, etc...

Además, hemos reforzado nuestra habilidad con el soldador, ya que hemos necesitado soldar en un espacio reducido los componentes, ya por si reducidos, con la complejidad que esto conlleva.

En cuanto a los problemas a los que nos hemos enfrentado a lo largo del desarrollo de esta fase, debemos destacar dos problemas principales.

En primer lugar, el Reset de nuestra placa. Este fue uno de los principales problemas que nos surgieron durante esta fase. Después de realizar el primer desarrollo de nuestro esquemático, enrutar las pistas, generar los archivos gerber y de taladros y mandar a fabricar la placa, tras soldar el sensor y el microcontrolador y conectarlos al ordenador, nos dimos cuenta que el ordenador no era capaz de reconocer la placa.

En una primera revisión partíamos del supuesto de que el error se centraba en un problema de soldaduras, ya que antes de realizarlas se realizaron pruebas de funcionalidad y no se reconoció problema alguno. Se decide entonces desoldar los componentes y revisar la PCB

por separado, así como las soldaduras de los pines de ambos dispositivos de forma independiente, sin conseguir llegar a resultado alguno.

Es en una segunda revisión cuando se detecta el problema, determinándose que proveniente del diseño del primer esquemático. El Reset de la placa Pro-Micro es activo bajo, y en nuestro esquemático la habíamos considerado activo alto, por tanto, habíamos generado un reset continuo en el circuito, y por ello nuestra placa no era reconocida por PC.

Para asegurarnos de que este era realmente el problema, conectamos otra placa Pro-Micro al PC uniendo el reset mediante una conexión por cable a la tierra y comprobamos que efectivamente el ordenador no la reconocía, mientras que según quitábamos dicha conexión el ordenador detectaba la placa al instante.

Descubierto el error, tuvimos que cambiar dicho reset y quitarle la pista que lo unía a la tierra, y mandar de nuevo a fabricar nuestra placa.

En segundo lugar, tuvimos problemas a la hora de realizar las soldaduras. Debido a la poca disponibilidad horaria de la que disponía, decidí utilizar mi propio soldador para realizar las últimas soldaduras.

Dicho soldador es significativamente más grueso que el de los laboratorios, por lo que en algunos pines la soldadura fue demasiado ancha, saliéndose del incluso del pin. Se tuvo entonces que desoldar los pines que estaban más soldados y arreglar las soldaduras incorrectas mediante la utilización de un desoldador.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

Tras la realización de este estudio podemos sacar múltiples conclusiones ateniéndonos a temáticas específicas que giran en torno situaciones diversas y, abarcando aspectos que atañen a la metodología, al conocimiento matemático e incluso a las habilidades prácticas, pero tres resultan fundamentales para definir el trabajo desarrollado.

En primer lugar, podemos decir que no siempre la obtención de los datos de un sensor, pese a lo que pudiera parecer a priori, es la tarea más laboriosa, sino que el posterior tratado, filtrado y la interpretación de los mismos puede suponer un desafío mucho mayor. Nuestra experiencia en el desarrollo ha dejado constancia de ello y el proceso de calibrado, filtrado y depurado de datos ha sido una tarea ardua, aunque al mismo tiempo enriquecedora por permitir ampliar nuestro conocimiento, aproximándolo a caos reales, donde la necesidad y la complejidad del tratamiento de los datos se ha convertido en una fase esencial de todo el proceso.

En segundo lugar, es destacable que el entramado matemático que esconden tanto los ángulos de Euler, pasando por la matriz de rotación y llegando especialmente a los

cuaterniones es bastante complejo, y su comprensión en profundidad, tal como se requiere para llegar a un dominio de las técnicas como el que se pretendía alcanzar con este trabajo, precisa de un manejo, de un dominio de conocimientos matemáticos que, aunque han alcanzado un nivel relativamente aceptables, aun distan de lo más deseable y se nos plantea como un reto para el desarrollo de futuros trabajos.

Por último podemos afirmar que el desarrollo de aplicaciones y dispositivos que utilicen sensores como las IMU son y serán una realidad debido al gran potencial que éstas nos ofrecen a la hora de describir el movimiento en tiempo real y este trabajo se convierte en una posible aportación de lo que pueden ser y representar estas tecnologías para la mejora de la calidad de vida de las personas.

5.2 Trabajo futuro

Aunque forma parte de una reflexión global, y en consecuencia podría formar parte del epígrafe anterior, lo cierto es que el desarrollo de este trabajo ha permitido asentar las bases de lo que podría definirse como mi propósito profesional a medio plazo.

Adentrarme en el mundo de los sensores y de su potencialidad han afianzado mi perfil de especialización y mi futuro ha quedado marcado y orientado en torno a estos dispositivos y otros similares cuya potencia radica en las posibilidades de medición en tiempo real como un instrumento con versatilidad de aplicaciones en múltiples campos.

Adicionalmente a este planteamiento general, en un futuro inmediato y con relación a los fundamentos matemáticos, considero que se ha abierto y puesto de manifiesto la necesidad de profundizar en el estudio y la comprensión en detalle de los cuaterniones como método para representar el movimiento, con el fin de optimizar los algoritmos que nos permiten calcularlo.

En este afán de ampliar conocimientos, pero siempre bajo la perspectiva de crear instrumentos y mecanismos útiles para la mejora social y personal, también se plantean como desarrollos de futuro:

Con respecto al hardware:

- Reducir las dimensiones, eliminando los PCB tanto del sensor como del procesador y diseñando un nuevo PCB que además incluya
 - Alimentación
 - Un CI para Wireless
 - Un sistema para programar el micro

Con respecto al software:

- Implementar el uso de cuaterniones para acelerar y mejorar la obtención de los datos sensados
- Diseñar la aplicación de IA para el análisis de los datos, y la generación de alarmas para optimizar la postura corporal.

Referencias

- [1] L.H. Espinoza Páez. “Sistema para monitorear el grado de curvatura de la columna vertebral mediante la utilización de sensores IMU”. Tesis de grado de la autora, <http://dspace.esPOCH.edu.ec/handle/123456789/9002> (abril-2019).Chimborazo, Ecuador. 2017.
- [2] F.M. Kovacs, M.Gestoso, M.T.Gil Del Real, J. López, N. Mufraggi, J.I. Méndez. “Risk factors for non-specific low back pain in schoolchildren and their parents: a population based study”, Pain 103(3):259-268, Junio-2003.
- [3] R.Y.W. Lee “A real-time gyroscopic system for three-dimensional measurement of lumbar spine Motion”, Medical Engineering & Physics 25(10):817-824, Enero-2004
- [4] M.Miezal, G.Bleser, N.Schmitz, and D.Stricker. “A generic approach to inertial tracking of arbitrary kinematic chains”. International Conference on Body Area Networks, Boston, MA, USA, 2013.
- [5] T.Seel, J.Raisch, T.Schauer “IMU-Based Joint Angle Measurement for Gait Analysis”, ISSN 1424-8220, Sensors(14):6891-6909, 2014
- [6] C.Antonya, S. Butnariu “Real-time Representation of the Human Spine with Absolute Orientation Sensors”, 2016 14th International Conference on Control, Automation, Robotics & Vision, Phuket, Thailand, 13-15th Noviembre-2016.
- [7] M.Kraft, “Micromachined Inertial Sensors State of the Art and a Look into the Future”, University of Southampton, Measurement+Control(33), Julio-2000.
- [8] R.K.Curey, M.E.Ash, L.O.Thielman, C.H.Barker, "Proposed IEEE inertial systems terminology standard and other inertial sensor standards" Position Location and Navigation Symposium, PLANS: 83-90, Abril-2004.
- [9] A.Manzanares del Moral, A.Luque, “Estudio de Modelos matemáticos de Acelerómetros Comerciales”, Tesis de Máster del Autor, Universidad de Sevilla, España, 2008
- [10] N.Yazdi, F.Ayazi, K.Najafi, “Micromachined Inertial Sensors”, Proceedings of the IEE, 86(8):1640-1659, Agosto-1998.
- [11] S.Zihajehzadeh, D.Loh, T.J.Lee, R.Hoskinson, E.J.Park ”A cascaded Kalman filter-based GPS/MEMS-IMU integration for sports applications“ Measurement, (73):200-210,Septiembre-2015, Elsevier.
- [12] S.Sasani, J.Asgari, A.R.Amiri-Simkooei “Improving MEMS-IMU/GPS integrated systems for land vehicle navigation applications” GPS solutions,20(1):89-100,Enero-2016.

- [13] S.Bose, A.K.Gupta, P.Handel “On the noise and power performance of a shoe-mounted multi-IMU inertial positioning system” International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, Septiembre-2017.
- [14] Z.Liab, G.Changa, J.Gaoa, J.Wanga, A.Hernandez “GPS/UWB/MEMS-IMU tightly coupled navigation with improved robust Kalman filter” Advances in Space Research 58(11): 2424-2434, Diciembre-2016.
- [15] Obtenido de: <https://learn.sparkfun.com/tutorials/accelerometer-basics/all> a fecha 18/01/2019.
- [16] H.Cuevas “Diseño de un sistema de vigilancia costera y salvamento marítimo basado en la cooperación entre plataformas autónomas aéreas y navales” Tesis de fin de Máster del autor, Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación, 2018.
- [17] Extraído de <https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards> a fecha de 10/12/2018
- [18] E.Serrano,R.O.Sirne,G.la Mura “Rotaciones, secuencia aeroespacial y cuaterniones. Una revisión de las relaciones fundamentales” Ciencia y Tecnología, ISSN 1850-0870 (14): 11-28, 2014.
- [19] R. Vázquez Valenzuela “Cinemática y Dinámica de la Actitud” Escuela Superior de Ingenieros, Universidad de Sevilla, Mayo 2012.
- [20] Extraído de <https://www.luisllamas.es/> a fecha de 19/02/2019
- [21] T.Fischer, M.A.Nitsche, S.Pedre “Fusión de encoders de cuadratura, sensores inerciales y magnéticos para la localización de robots móviles”, Laboratorio de Robótica y Sistemas Embebidos, Departamento de Computación, Marzo-2014.
- [22] R.Mahony, T.Hamel “Complementary filter design on the special orthogonal group” Decision and Control, and European Control Conference, 2005.
- [23] R.Mahony, T.Hamel “Nonlinear Complementary Filters on the Special Orthogonal Group” IEE Transactions on Automatic Control, Junio-2008.
- [24] S.O.H. Madgwick “An efficient orientation filters for inertial and inertial/magnetic sensor arrays” Abril-2010.
- [25] Extraído de <https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/all> 10/03/2019.

Anexos

A Ángulos de Euler vs cuaterniones

En este anexo se profundizará en las relaciones matemáticas que existen entre los Ángulos de Euler y los cuaterniones. Se estudiarán las ventajas de utilizar unos u otros así como los inconvenientes y problemas que pueden surgir a la hora de calcularlos. Para relacionar ambas partes utilizaremos la matriz de rotación, la cual nos permitirá describir el movimiento de rotación de un objeto en el espacio de tres dimensiones.

Comenzaremos por estudiar las fórmulas clásicas que relacionan los elementos citados anteriormente. A pesar de que estas fórmulas ya han sido estudiadas y analizadas en profundidad en otros estudios, su interpretación y justificación no son del todo claras, lo que resulta en una gran dificultad a la hora de implementar algoritmos o programas que hagan uso de ellas [18]

Sea u un vector $\neq 0$ definido en el espacio de R^3 . Se pretende calcular el vector u' vector que surge de la rotación de u alrededor de un vector v un ángulo θ en sentido positivo. Según [18], mediante el desarrollo de operaciones vectoriales, se puede llegar a definir esta rotación como:

$$u_{rot} = \frac{[\cos \theta + \sin \theta (\omega) + [1 - \cos \theta](v)(v)^r]}{R_\theta}(u)$$

Ecuación 1: Rotación de un vector obtenida de [18].

Esta es ecuación conocida como Fórmula de Rodrigues-Euler en forma matricial, donde podemos definir R_θ como la matriz de rotación. Si desarrollamos esta ecuación matricial podemos expresar dicha matriz de la siguiente forma:

$$R_\theta = \begin{pmatrix} [1 - \cos \theta]v_1^2 + \cos \theta & [1 - \cos \theta]v_1v_2 - \sin \theta v_3 & [1 - \cos \theta]v_1v_3 + \sin \theta v_2 \\ [1 - \cos \theta]v_1v_2 + \sin \theta v_3 & [1 - \cos \theta]v_2^2 + \cos \theta & [1 - \cos \theta]v_2v_3 - \sin \theta v_1 \\ [1 - \cos \theta]v_1v_3 - \sin \theta v_2 & [1 - \cos \theta]v_2v_3 + \sin \theta v_1 & [1 - \cos \theta]v_3^2 + \cos \theta \end{pmatrix}$$

Ecuación 2: Matriz de rotación desarrollada obtenida de [18].

Ahora que hemos definido la rotación de un vector mediante su matriz de rotación podemos empezar a definir y estudiar los cuaterniones.

En primer lugar definimos un cuaternión como una extensión de los números complejos a 4 dimensiones. Definidos por Hamilton en el siglo XIX fueron concebidos por el mismo como un lenguaje universal para la física, aunque pronto fueron sustituidos por vectores y matrices [19].Escribiremos por tanto un cuaternión como:

$$q = q_0I + iq_1 + jq_2 + kq_3$$

Ecuación 3: Definición de un cuaternión [19]

Equivalentemente podemos expresar nuestro cuaternión q como:

$$q = \begin{pmatrix} a & -b & d & -c \\ b & a & -c & -d \\ -d & c & a & -b \\ c & d & b & a \end{pmatrix}$$

Ecuación 4: Ecuación matricial del cuaternión q obtenida de [18].

Donde

$a = q_0$, $b = q_1$, $c = q_2$ y $d = q_3$.

Ahora bien, ¿Cómo podemos relacionar la matriz de rotación calculada anteriormente con los cuaterniones?

La respuesta está en la Fórmula de Rodrigues-Euler mencionada anteriormente junto con la definición del producto de cuaterniones, descrito en [18]. Siguiendo el desarrollo matemático explicado con detalle en la referencia anterior llegamos a la siguiente expresión de la matriz de rotación expresada en función de los cuaterniones a , b , c y d .

$$R_\theta = \begin{pmatrix} 2a^2 + 2b^2 - 1 & 2(bc - ad) & 2(ac + bd) \\ 2(bc + ad) & 2a^2 + 2c^2 - 1 & 2(-ab + cd) \\ 2(-ac + bd) & 2(ab + cd) & 2a^2 + 2d^2 - 1 \end{pmatrix}$$

Ecuación 5: Matriz de rotación mediante cuaterniones [18].

Adicionalmente se puede demostrar que la rotación de un vector $u \in R_3$ alrededor de un eje orientado por v un ángulo θ , puede representarse mediante un cuaternión unitario q definido por la siguiente expresión:

$$\bar{q} = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} \bar{v}$$

Ecuación 6: Rotación de un vector mediante cuaterniones obtenida de [19].

Una vez hemos definido qué son los cuaterniones es momento de definir que son los ángulos de Euler y cómo están relacionados con la matriz de rotación y los cuaterniones.

En general podemos decir que la orientación de un cuerpo en el espacio tridimensional puede definirse mediante tres rotaciones sobre ejes no consecutivos. Dado que el sistema de representación de la realidad es un espacio tridimensional y hacen falta tres rotaciones para definir la orientación de un cuerpo, podemos fácilmente obtener 12 posibles rotaciones para definir esta orientación.

Tradicionalmente en aeronáutica se ha conocido a estas tres rotaciones como ángulos de Euler. En la siguiente figura podemos observar estas tres rotaciones más claramente:

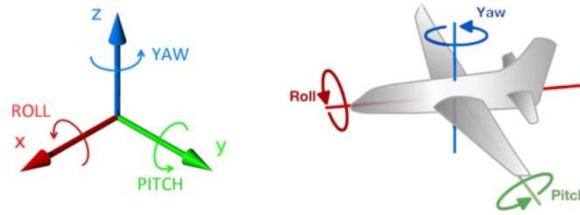


Figura que representa los ángulos de Euler obtenida de [20].

Como se ha visto anteriormente, para definir la orientación de un cuerpo, habría que realizar tres rotaciones de ejes no consecutivos. Si como se ha definido, cada rotación alrededor de un vector viene dada por una matriz de rotación, para obtener la orientación de un cuerpo habría que realizar el producto de tres matrices. Al no ser conmutativo este producto y como se ha mencionado anteriormente podríamos encontrar hasta doce secuencias distintas que definirían la orientación de nuestro cuerpo bajo estudio.

A cada uno de los tres elementos que forman estas doce secuencias de rotación lo llamaremos ángulo de Euler. A partir de esta definición y como se puede ver más detalladamente en [19] se llega a una relación de estos ángulos con la matriz de rotación.

Por último para relacionar los cuaterniones con los ángulos de Euler descritos anteriormente nos basaremos la ecuación 6. De esta forma obtenemos una correspondencia directa para cada ángulo de Euler con un cuaternión de la forma:

$$q_{\omega} = \cos \frac{\omega}{2} + k \sin \frac{\omega}{2} \quad \text{Donde } q_{\omega} \text{ es el ángulo de Euler } (\psi, 0, 0)$$

$$q_{\theta} = \cos \frac{\theta}{2} + j \sin \frac{\theta}{2} \quad \text{Donde } q_{\theta} \text{ es el ángulo de Euler } (0, \theta, 0)$$

$$q_{\phi} = \cos \frac{\phi}{2} + i \sin \frac{\phi}{2} \quad \text{Donde } q_{\phi} \text{ es el ángulo de Euler } (0, 0, \phi)$$

Ecuaciones de los ángulos de Euler en función de cuaterniones obtenidas de [19].

En nuestro caso de estudio nos decantaremos por la utilización de los ángulos de Euler en detrimento de los cuaterniones debido a dos factores principalmente. Por un lado los ángulos de Euler pueden definirse en el espacio de los números reales, mientras que los cuaterniones son extensiones de los números complejos. Por otro lado la representación de ángulos de Euler es mucho más intuitiva y visual. Sin embargo al seleccionar esta forma para la representación de los datos deberemos tratar con algunos problemas, principalmente lo que se conoce como “Gimbal Lock”, que supone la pérdida de un grado de libertad debida a la posible alineación de dos ejes de rotación, y el problema de la deriva asociada al giróscopo que pese a no poder evitarse utilizando cuaterniones, si se reduciría su efecto notablemente.

B Filtro de Madgwick

En la actualidad existen numerosas técnicas de fusión e integración de los datos de aceleraciones obtenidos de sensores inerciales como el acelerómetro y el giróscopo. Si bien es cierto que dependiendo de la aplicación en concreto habrá que adaptar estas técnicas según las necesidades de la misma, en la actualidad las técnicas de integración más extendidas son el Filtro de Kalman y el filtro complementario.

Podríamos decir que el filtro de Kalman es una de las técnicas de fusión más utilizadas debido a su gran versatilidad y eficacia. Nos permite adaptarlo de manera relativamente sencilla a las necesidades de la aplicación. Sin embargo este tipo de filtros, presentan ciertas desventajas en algunos tipos de aplicaciones, como podría ser en nuestro caso de estudio.

En primer lugar estos filtros requieren definir un modelo probabilístico para los datos que queremos integrar. Por otro lado para poder aplicar esta técnica de filtrado debemos asumir que las ecuaciones del problema en cuestión son, en su totalidad, lineales. Es por esta última razón y debido a la no linealidad de la cinemática rotacional de nuestro problema por lo que no utilizaremos este tipo de filtros en nuestra aplicación [21].

En nuestro caso la técnica que decidimos implementar fue el filtro de Madgwick. Este filtro surge a partir de un sistema ideado por Mahony, en el cual se aprovecha la parametrización del espacio de rotaciones a través de cuaterniones [22] para implementar un filtro complementario, incorporando una estimación del sesgo del giróscopo. A pesar de que esta técnica mejora notablemente los resultados del filtro de Kalman para aplicaciones no linealizables, aún poseía un error constante en la estimación de las medidas debido la falta de un observador absoluto ortogonal [23].

La matemática detrás de este filtro, así como los procedimientos que utiliza para compensar la deriva del giróscopo, la distorsión del magnetómetro, o la implementación de la ganancia del mismo quedan fuera de este trabajo. Para obtener más detalles sobre este tipo de filtro, sus algoritmos para la fusión de los datos...etc consultar [24].

En nuestro caso concreto utilizaremos la implementación de un filtro Madgwick en lenguaje C y la adaptaremos a nuestras necesidades.

C Código Arduino utilizado:

```
#include <MadgwickAHRS.h>
#include <IMU_MPU9250.h>

float roll;
float pitch;
float yaw;
long microsPerReading;
long Serialtime;
long last_time;

Madgwick filter;
MPU9250 imu;

void updateIMU(float *roll, float *pitch, float *yaw){

    if (imu.readByte(MPU9250_ADDRESS, INT_STATUS) && 0x01){
        imu.readAccelData(imu.accelCount);
        imu.getAres();

        imu.ax = (float)imu.accelCount[0]*imu.aRes;
        imu.ay = (float)imu.accelCount[1]*imu.aRes;
        imu.az = (float)imu.accelCount[2]*imu.aRes;
        imu.readGyroData(imu.gyroCount);
        imu.getGres();

        imu.gx = (float)imu.gyroCount[0]*imu.gRes;
        imu.gy = (float)imu.gyroCount[1]*imu.gRes;
        imu.gz = (float)imu.gyroCount[2]*imu.gRes;
    }

    imu.updateTime();
    imu.delt_t = millis() - imu.count;
    if (imu.delt_t > microsPerReading){
        filter.updateIMU(imu.gx, imu.gy, imu.gz, imu.ax, imu.ay,
imu.az);
        *roll = filter.getRoll();
        *pitch = filter.getPitch();
        *yaw = filter.getYaw();
        imu.count = millis();
    }
}

void setup() {
    Wire.begin();
    Serial.begin(9600);
    filter.begin(25);

    microsPerReading = 1000 / 25;
    Serialtime = 50;
    last_time = millis();
    imu.calibrateMPU9250 (imu.gyroBias, imu.accelBias);
    imu.initMPU9250 ();
}
```



```
void loop() {  
  updateIMU(&roll, &pitch, &yaw);  
  if (millis() - last_time > Serialtime) {  
    Serial.print("Ori ");  
    Serial.print(yaw);  
    Serial.print(" ");  
    Serial.print(pitch);  
    Serial.print(" ");  
    Serial.println(roll);  
    last_time = millis();  
  }  
}
```

D Código matlab utilizado para la representación de los ángulos de Euler:

```
%Script que representa los ángulos de Euler segun el movimiento de
nuestros sensor
close all;
clc;
y=zeros(1,10000);
puerto_serial=serial('COM6');
fopen(puerto_serial);
yaw=zeros(1,500);
pitch=zeros(1,500);
roll=zeros(1,500);
a=zeros(1,500);
numero_muestras = 500;
contador_muestras = 1;
inicial = 0;
n=1:numero_muestras;

for i=0:numero_muestras
    valor_angulos = fscanf(puerto_serial,'%f %f %f');
    if(inicial == 0)
        pause(5);
    end
    yaw(contador_muestras) = valor_angulos(1,1);
    pitch(contador_muestras) = valor_angulos(2,1);
    roll(contador_muestras) = valor_angulos(3,1);
    contador_muestras = contador_muestras +1;
    inicial = 1;

disp(valor_angulos);
end

figure('Name','Angulos Euler')
title('Euler angles');
xlabel('Muestras');
ylabel('Grados');
grid on;
hold on;
plot(n,yaw(1:500));
grid on;
hold on;
plot(n,pitch(1:500),'g');
grid on;
hold on;
plot(n,roll(1:500),'r');
legend('Yaw','Pitch','Roll');

fclose(puerto_serial);
delete(puerto_serial);
clear all;
```